# React 学习笔记

### React全家桶+Antd 视频

1. **React环境搭建（第三章）**

配置less-loader

{

loader: require.**resolve**('less-loader'),

options:{

**javascriptEnabled: true, // 如果less版本是3.X以上的，添加这句话**

modifyVars: {

"@primary-color":"#f9c700" // 自定义 antd主题

}

}

}

1. **React基础知识（第二章）**

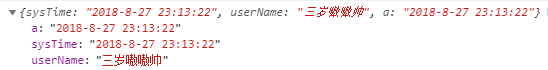
**第三课：**

注意两种函数声明的时，this的区别



1. **States的用法**

****





// 刚开始学的时候没有弄清楚state = { } 的用法，在html标签中引用时，如果没有在state = { } 中初始化，那么页面加载时这个数据会在若干秒后才显示

1. **React-Router（-dom）**

个人理解：

react-router-dom 是 react-router 的更高级版。

**exact**: 精准匹配

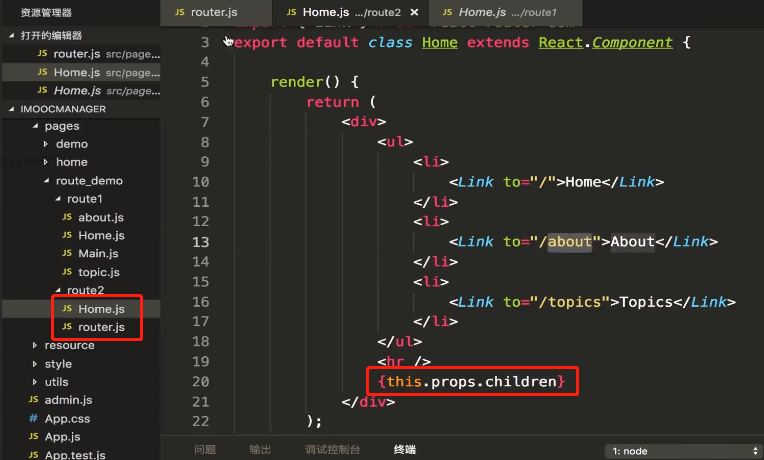
**/:location**匹配所有路由

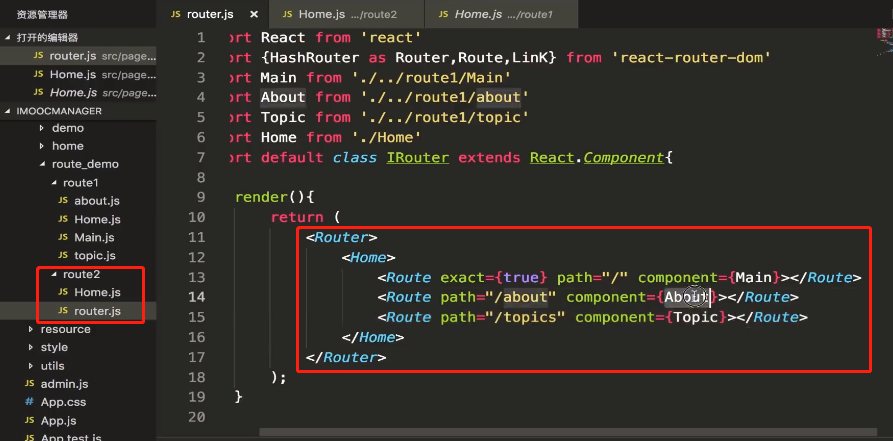
<Route path=“/:location” component={ XX }> </Route>

（方法一：）



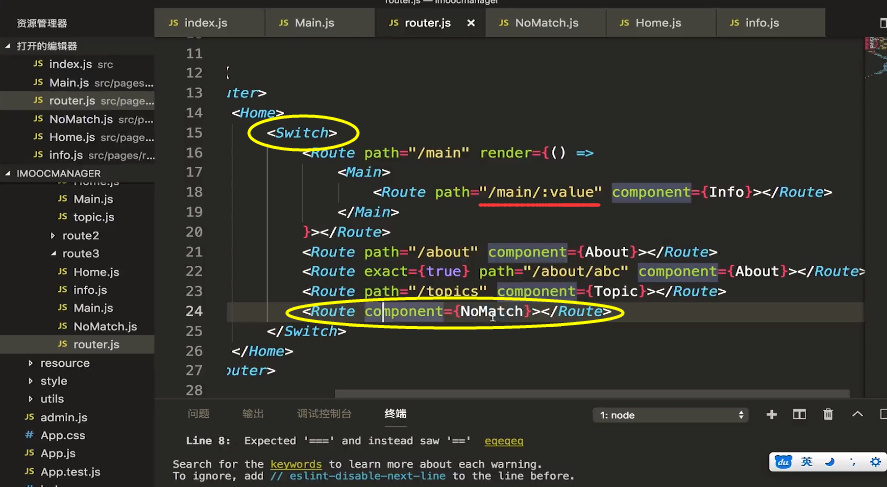
（方法二:）





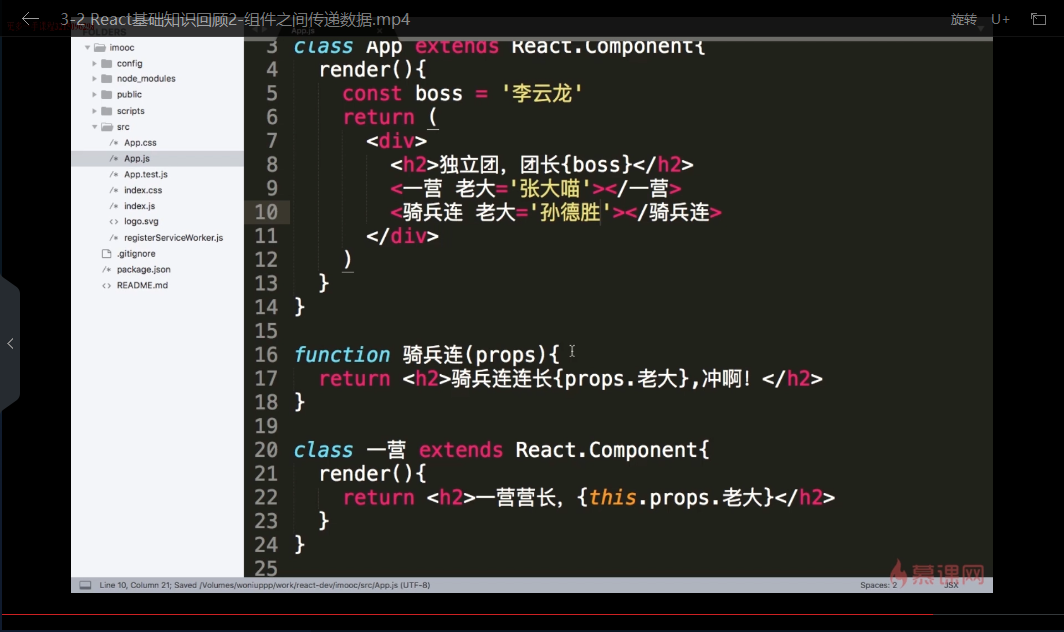
（方法三：）

Switch只匹配命中的第一个路由



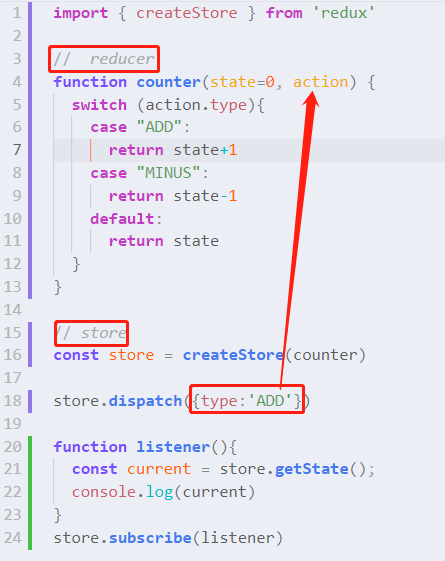
### React+redux+express开发招聘网站

**一、React组件之间的数据传递（第三章）**



**二、Redux状态管理与React-router（第四章）**

redux的基本使用



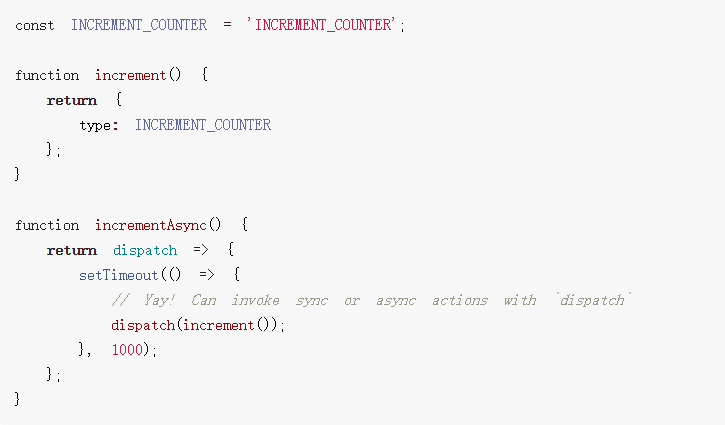
**redux异步**

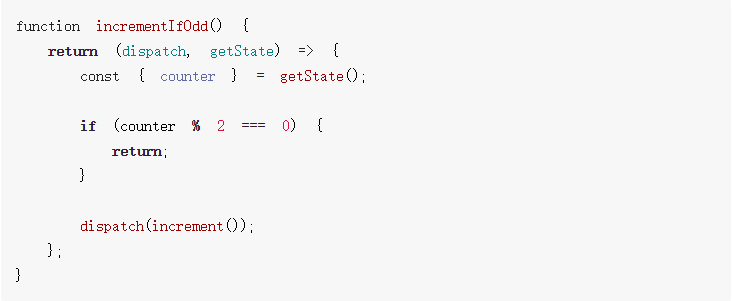
npm install redux-thunk –save

使用applyMiddleware开启thunk中间件

import { appluMiddleware } from ‘redux’

import { thunk } from ‘redux-thunk’

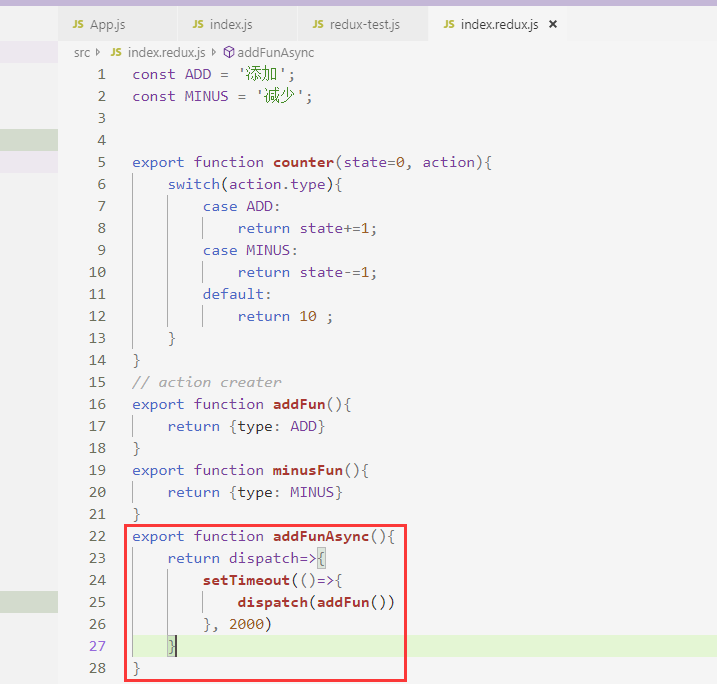




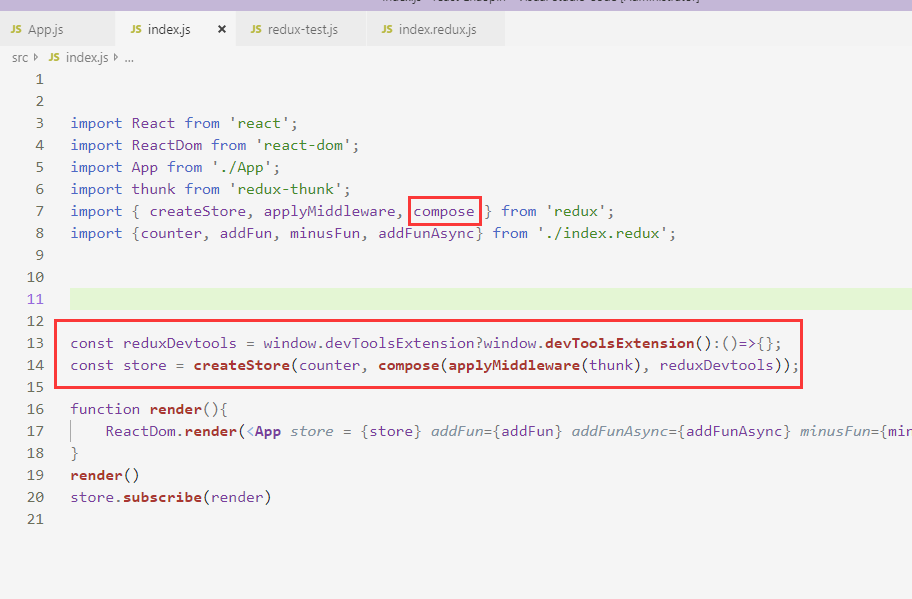
4.4（02：00）

redux处理异步： 中间件redux-thunk





redux 调试插件 ReduxDev Tools:



redux-router

Provider 组件在应用最外层，传入store即可，只用一次

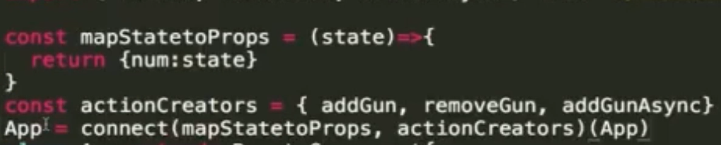
Connect 负责从外部获取组件需要的参数

**装饰器**

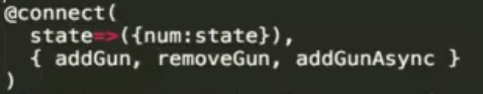
**npm run eject**

**npm install babel-plugin-transform-decorators-legacy**

用装饰器之前：



用装饰器之后：



**使用Axios发送异步数据**

端口不一致，使用proxy配置转发

axios拦截器，统一处理loading

axios.interceptors.request.use(function(config){

return config

})

axios.interceptors.response.use(function(config){

return config

})

**1.组件引用图片**

**方法一：** const divStyle = {

background: `url(${require("./a1.png")})`

};

style={ divStyle }

**方法二：** import A from './a1.png';

**2.子组件定义了alert()事件，在父组件引用时触发子组件中的alert()事件。**

**子组件中：**<div onClick={ this.props.yell }>温馨提示</div>

**父组件中：**

yellFa= () => {

this.refs.father.yell();

}

<Login yell={this.yellFa} ref="father" />

**3.同一个组件不同的样式:**

render(){

var sty = {

color: this.props.fontColor,

}

return(

<div style={sty} fontColor="#333" ></div>

)

}

**4.**

**getInitialState:** 组件被浏览器加载之前调用【ES6中不需要写，写在constructor中，this.state = {a: "0"} 】

**componentDidMount:** 在组件被浏览器加载之前，但是组件的render还没被调用之前触发

**样式融合，**var textStyle = {

emphasis:{

fontSize: 10,

...titStyle

}

}